

Opening Windows and Frames

Opening a new window is easy. Use the **open()** method. However, this isn't the same **open()** method we used for streams. This open method takes two or three variables. The syntax is:

```
[window_var=] [window].open("URL", "window_name", ["window_features"])
```

The items in brackets are optional.

This first variable is the URL. Just give it a URL string. Pretty easy. Next comes the name of the window. This is a unique identifier of the name of the window. If you specify the name of a window that does not exist, one will be created. If you specify one that does exist, then the URL is sent to that window.

The last part is optional. Here you can specify some special attributes for the window. If you don't specify any attributes, then the window looks like a normal web browser. However, you can set several features of it explicitly. You can change such things as the size, whether or not there is a menubar, etc.

Example:

```
<SCRIPT LANGUAGE="JavaScript">
  my_new_window=window.open("", "", "menubar=no");
  my_new_window.document.write("This is a new window.");
  my_new_window.document.close();
</SCRIPT>
```

This will pop up a new web browser without a menubar and print some text to it.

The *window_features* command can take several options.

- toolbar [=yes|no] [=1|0]
- location [=yes|no] [=1|0]
- directories [=yes|no] [=1|0]
- status [=yes|no] [=1|0]
- menubar [=yes|no] [=1|0]
- scrollbars [=yes|no] [=1|0]
- resizable [=yes|no] [=1|0]
- width=*pixels*
- height=*pixels*

To specify multiple options, separate them by commas. The boolean value on the right side can be specified in several ways. The following examples have the exact same effect:

```
toolbar=yes
toolbar=1
toolbar
```

As you can see, giving no boolean value is the same as **yes**.

If you wish to close a window, use the **close()** method. In our previous example, we opened a window called **my_new_window**. If we wish to close it, use:

```
my_new_window.close();
```

- Pretty simple, eh? If you wish to close the current window, you can use either:

```
window.close();
```

or

```
self.close();
```

Frames

To create some frames, you need to first create a window. Once you have done that, you need to send the appropriate HTML to that window to create the frames. It'll look a little like this:

```
some_new_window=open("", "");
some_new_window.document.write("<html><frameset cols=\"*,*\">>\n",
                                "<frame src=\"frame1.html\" name=\"fram",
                                "<frame src=\"frame2.html\" name=\"fram",
                                "</frameset></html>");
some_new_window.document.close();
```

Now, if you wish to send data to the appropriate frame, you can do:

```
some_new_window.frame1.document.open();
some_new_window.frame1.document.write("I am now writing in frame1.");
some_new_window.frame1.document.close();
```

If you wish to close the two frames you just created, you can just write to the parent document. When you call the **open()** method, this will clear the parent window and remove the **<frameset>** tags. The result is that your frames will be clobbered. Example:

```
some_new_window.document.open();
some_new_window.document.write("This is a new document without frames");
some_new_window.document.close();
```

The above code just cleared the window (when we called **open()**) and wrote some text to it, thus removing the frames.

Dialog and Alert Boxes

Dialog and alert boxes are usefull for a variety of purposes. You can use them to get input from the user. You can also use it for debugging purposes.

First, the **alert** box. The alert box pops up in the middle of the screen with some text. It gives the user an OK box. Example:

```
alert("That is an invalid value.");
```

That's it.

The next item is the **confirm** dialog box. The confirm dialog box is the same as the alert except that it

- gives an **OK** and **Cancel** buttons. Then, it returns true or false corresponding to OK or Cancel respectively. Example:

```
if(confirm("Would you like to try again?")) {  
    document.location="index.html";  
}  
else {  
    quit();  
}
```

The final type of dialog box is the **prompt** dialog box. The prompt dialog box is the same as the confirm dialog box, but the user can input some data. The `prompt()` command then returns the value that the user entered. You can optionally specify the default value. The format is:

```
prompt(message [, default_value] )
```

Example:

```
username=prompt("Please enter your name.", "");
```

Another example:

```
the_value=prompt("Please enter the amount you wish to bet.",0);
```

It is always recommended that you give it a default value. If you do not specify a default value, then the string "undefined" is placed in the input box which looks really bad.

Return to the [Front Page](#).
Go to the next lesson: [Event Handling](#).

HyTime International Conference
Seattle, WA,
20-21 August 1996

Catherine Hamon <- Michel Biezunski -> Michel Biezunski, High Text, S.A.R.L.
Catherine Hamon, High Text, S.A.R.L.

Topic Map A Topic Map of This Conference's Proceedings

Why does a Topic Map fit Conference Proceedings?

The purpose of a Topic Map-based hyperdocument is to interconnect semantically heterogeneous information. Conference Proceedings seemed to us to be a good sample of a type of hyperdocument that is adapted to a Topic Map.

<- Anchor -> A Topic Map allows readers to navigate following topics that can appear in multiple documents. Rather than just being a simple term, a topic is a link that contains a title and is pointing to places in the documents where there are occurrences of this topic. These places, otherwise called anchors, can be grouped following various roles they play, and the anchor roles orient the navigation (e.g., definition, mention, example, etc.).

Multi-document Indexes A Topic Map is fonctionnally equivalent to multi-document indexes, glossaries, and thesauri. Topics are organized in types, each instance of a topic type has a title, and each occurrence of a given topic in a document is described including the semantics of the anchor role.

A HyTime-based HTML hyperdocument

<- HTML -> The current hyperdocument is encoded in HTML. Therefore it can be browsed using one's favorite Web browser. Even the table of contents and the different indexes are a set of HTML links allowing to navigate to any appropriate location in the hyperdocument.

<- Topic Navigation Maps -> The architecture on which the hyperdocument is based conforms to the HyTime-based Topic Navigation Map specification. This specification, as it stands today, is being proposed as a committee draft for an ISO standard (ISO/IEC 13250). The architecture has been designed by the CAPH Committee, chaired by Steven R. Newcomb. We have played an active role in the design of the architecture. Extensions and fixes are being proposed and will be discussed before it will be completed as an ISO standard.

<- Ilink -> The idea of using semantics to help users find their way through the maze of links originates from HyTime, and more precisely from the ilink architectural form, because ilinks have a type (element type) and force semantics to be defined for each anchor or anchor aggregate (anchrole attribute).

EnLIGHTeN A Topic Map created by computer

High Text, SARL, is developing since the beginning of 1995 an application, called EnLIGHTeN, that

provides an environment to help create, maintain and navigate topic maps.

EnLIGHTeN is an application-specific way to deal with Topic Maps. It allows its users to index documents (either by hand or automatically or any combination of both), by declaring a list of topics. Each time a hit is found, a mark is inserted in the source documents. The current syntax we use is made of specific SGML comments (which really should be considered as processing instructions). The reason for this choice is to allow SGML users to insert topic marks in any document, without having to alter their existing DTDs and applications.

EnLIGHTeN accepts 3 kinds of source documents, the three types can be mixed in the same hyperdocument:

1. Plain vanilla ASCII
2. HTML (any version up to 3.2). The current version of EnLIGHTeN is limited to SGML-like documents (No DOCTYPE required, but tags should not be overlapping)!
3. SGML (any DTD)

EnLIGHTeN allows users to declare any type of relationships between topics, that are declared using a specific syntax. These relationships can have any semantics, but are limited to one-to-one relationship, in the current version. When relationships are one-to-many, they are decomposed in a set of one-to-one.

<- Document Type Definition -> Topic types and anchor roles are freely defined by users as parts of each index mark. EnLIGHTeN is a batch process which creates the HyTime Topic Map and the various HTML output documents from what it collects in the source documents. This is possible because the HyTime DTD and instance for the Topic Map are built on the fly. In the current version, the Topic Map DTD and instance are destroyed and re-created each time the hyperdocument is processed. We plan to use them as "control centers" of the connections within various integrated database-document environments.

Navigating an EnLIGHTeN-created hyperdocument

<- WWW -> The current version of EnLIGHTeN outputs HTML format, to allow users to publish directly their documents on the Web, and take advantage of existing Web browsers. Any HTML-based browser can be used. No CGI, Java, or specific Web browsers are involved. The HTML output is parsable using the HTML 3.2 DTD (dated May, 1996).

If the source documents were in HTML, and if they included graphics, sounds, video, special effects, ..., all those features are preserved and will still work in the output Topic Mapped-HTML documents.

EnLIGHTeN-created screens

The table of contents (**Contents**) is simply the list of the documents that are present in the hyperdocument. Clicking on any of them will open the document.

The indexes (**Indexes**) contains a list of topic types (i.e. persons, concepts, organizations, etc.). Clicking on one of them will open a given index, which is an alphabetic list of topics.

The topic screens are built by EnLIGHTeN by grouping all relevant links or anchors pointing from a given topic. These screens are divided into two parts

- <- Relationship -> Related topics: gives the list of all topics related to the current one, with the specific semantics of each link.

- Occurrences: gives the list of occurrences, sorted by anchor roles. Each line contains the beginning of the content of each occurrence, allowing users to have supplementary information and to decide which occurrence is the most interesting to traverse to.

<- Anchor -> The documents are displayed with anchors, that let users know which topic instances are anchored at a given location. Clicking on each topic instance triggers the display of the corresponding topic screen.

When there are more than one occurrence of a given topic with a given anchor role, then a pair of arrows [$<-$] and [$->$] are displayed around the topic instance. They allow users to traverse directly to the next (or previous) occurrence of the same topic, whether it is located in the same or in another document.

The "info" option displays a report on the current state of the topic map. It gives information about:

- the number of documents present in the hyperdocument,
- the list of topic types used, each of which followed by the list of corresponding anchor roles for specifying occurrences
- the list of topic relation types used, each of which followed by the list of corresponding anchor roles for specifying the two ends of the relationship,
- information about the instance: each topic type is followed by a list of the topic instances belonging to it, followed by the total number of occurrences in the documents. When a topic instance is related to others, the relationships and the other topics related are listed under it.

Trick: while you are reading a document, whenever you want to see either the table of contents or an index, click on any topic, and the menu [Contents|Indexes|Info] is always displayed at the bottom of the screen for each topic.

How this hyperdocument was created?

We have asked each of the contributors to the HyTime Conference to give us a copy of their paper, in electronic form. Most of them were in ASCII, some of them were in HTML, one of them was in Microsoft Word and was converted into HTML. The graphics have been converted to GIF or JPEG format, to ensure compatibility with the popular Web browsers.

Each author was able to insert his or her own topics as marks (SGML comments) into their documents, or alternatively to provide us with a list of topic that were considered relevant to the domain. We have added our own list.

We have used EnLIGHTeN's Ascii to HTML feature to transform ASCII documents into HTML. Some formatting has been rebuilt "by hand".

We have used EnLIGHTeN's automatic indexing feature to insert into all documents the topics that have been declared into some of them.

The first version of this topic map was presented in Seattle, at the end of the conference. We received about half of the documents a few days before the conference started, and the rest the first day of the conference, i.e. just one day before the presentation.

It takes us about 5 minutes to build the current HTML set of Topic Mapped documents with a 486-75 computer (Linux), and approximately the same time on a Sun-Sparc 20 Unix workstation. In Seattle, time was about 45 minutes. The code of EnLIGHTeN has been considerably rewritten since.

In order to minimize the intellectual work required by indexing and choosing meaningful anchor roles,

we decided that the standard basic anchrole was the first name of the author, in the scope of his/her own document. Thus, for example, the topics present in David's paper appeared on the screen as a list under the name of David. This allowed us to see for several topics, that they were mentioned by different authors. But this design was far from being ideal, because it creates a confusion between the document source and the anchor role.

The current version of this topic map has been redone, without any change to the documents themselves. We have worked on several steps.

- Delete unwanted occurrences produced by automatic indexing
- Create and differentiate anchor roles. For "authors", we have created the following list of anchor roles:
 - abstract: abstract of the paper in the conference's program.
 - program: places where the author appears in the conference's program.
 - paper: points to the actual paper (if available).
 - mention: occurrences where an author is mentioned by another one.
 - biography: biographical description of the author (appears only once here, for the keynote speaker, Len Bullard).
 - email: email address (if available).
- For "concepts", we have created the following list of anchor roles:
 - mention: when a concept is simply mentioned somewhere.
 - development: when a concept is fully developed in the part following a given location
 - definition: points to where a concept is defined (there can be several definitions of the same concept).

Notes about our design choices

There are a lot of arbitrary decisions that we had to take here, and we are taking all responsibility for the good (and bad) choices that have been done.

Note 1: We do not consider the documents presented here to be in a completely finalized form. We have discovered by doing this work that the Topic Map architecture is so open that there are several very different choices. We are almost sure that every one of the authors (or of the readers) will have a different point of view on how to describe the information. Also, we have probably forgotten very important topics, and have misused others. The nice thing about it is that we are able to make any changes to the whole structure very quickly. EnLIGHTeN will recalculate any possible change that someone will want to introduce. If you want to have something changed in this hyperdocument sample, by all means, don't hesitate to tell us. We plan to update this document every week (topic map part only) if necessary, to improve it.

It is also possible to declare multiple topic maps within the same set of hyperdocuments.

A topic mark can be considered a giant bookmark. Therefore a set of documents can also be marked by different users, each of whom defining his/her own topic map from the same set. A future version of EnLIGHTeN will include this feature, making it an SGML-based equivalent to annotation managers.

Note: In the currently designed HyTime Topic Map, each instance of an author element must have a fixed number of anchor roles, and therefore of anchor addresses. EnLIGHTeN has been designed in such a way that if an anchor role is not present, it will not be displayed, even if it is formally there. This remark has a general validity. EnLIGHTeN has been designed in a way that is useful for an end-user, and it doesn't simply reflect the HyTime structure. All these choices are not defined neither in the HyTime standard, nor in the Topic Map specification. They have been designed by High Text, and High Text bears the full responsibility for this interface. Another tool using Topic Maps could look and feel

very differently. This is why we feel it's interesting to have standards: future users of topic maps will (we hope) have the choice between different kinds of applications.

High Text's plans regarding EnLIGHTeN

<- Steven R. Newcomb -> GCA The Seattle Conference has been for us a "real world" live test experiment. We want to thank the organizers of the conference, namely Steven R. Newcomb and the GCA, to have allowed us to present this application, and the contributors to the conference, who all accepted to provide us with an electronic version of their paper. (The papers that are not included in this Topic Map have not been written in a deliverable form.)

We are now proposing a Topic Map Express Service, which is the equivalent of a printing service. Instead of giving the customer business cards, we deliver HTML topic maps. This enables us to keep our software at home, because we are not equipped today to offer needed facilities, including customer support, multiple versions, upgrades, documentations in different languages, etc.

This service is based on an initial consulting service (meetings, training, definition of an architecture specific to a given environment, ...), followed by a series of electronic document exchange: After you know how to encode your documents conforming to EnLIGHTeN's specification, you send us your documents, we process them on our machines, and send them back to you. If documents are liable to be sent through email, this could be a real quick service. Each time you need to have an update, we'll redo the processing and send it back. If documents should not be sent by email, we'll use fast-delivery international mail services.

High Text is now working to set up the conditions that will make possible to finalize EnLIGHTeN as a end-user product.

As a product, EnLIGHTeN will provide a bridge between databases and Internet/Intranet interfaces. EnLIGHTeN will be fed by documents stored in the database to automatically produce HTML pages according to a Topic Map.

Specific developments derived from the core technology are also possible, such as integration within different databases, alternative input or output formats.

Need more information? Send us email enlighten@hightext.com

Web Images Groups News Froogle [more »](#)

html parsable daterange:2449718-2451179

Search

[Advanced Search](#)
[Preferences](#)**Web**Results 1 - 10 of about 668 for **html parsable daterange:2449718-2451179**. (0.41 seconds)GDB Schema Documentation

GDB - Please read the Project Termination Announcement, Help: GDB Schema Documentation. ...

gdbwww.dkfz-heidelberg.de/gdb/schema.**html** - 9k - [Cached](#) - [Similar pages](#)HTML to the Max... In in our demo, and with **HTML**, this is done by pre-loading the ... keep the parsing code simple, one can restrict acceptable SGML to easily **parsable** forms without ...www.uic.edu/~cmsmcq/htmlmax.**html** - 36k - [Cached](#) - [Similar pages](#)Class java.lang.Integer

... Parameters: s - the String to be converted to an Integer Throws:

NumberFormatException If the String does not contain a **parsable** integer. ...tns-www.lcs.mit.edu/manuals/java-api-old/java.lang.Integer.**html** - 17k - [Cached](#) - [Similar pages](#)SoundCulture 96: Parsablesoundculture 96. **Parsable** Ron Kuivila. [when]. april 3 - 5. wednesday - friday:

12pm - 5pm. performance: friday, april 5; 8pm. [where]. the LAB. [what]. ...

www.lns.com/sc96/events/parse.**html** - 3k - [Cached](#) - [Similar pages](#)Michel Biezunski and Catherine Hamon - A Topic Map of This Any **HTML**-based browser can be used. No CGI, Java, or specific Web browsers are involved.The **HTML** output is **parsable** using the **HTML** 3.2 DTD (dated May, 1996). ...www.infoloom.com/IHC96/mb214.htm - 18k - [Cached](#) - [Similar pages](#)Using LiveHTML

... This means that LiveHTML parsed by that Listener will be able only to call static

HTML, environment variables, or other server **parsable** files, not executable ...kontti.helsinki.fi:8889/ows-adoc/lhtuserf.htm - 6k - [Cached](#) - [Similar pages](#)Conventions for HTML in Email... line above the quoted text, but no standard or machine **parsable** mechanism for this ...The overall goals for **HTML** Threading are: Enable sending UAs to provide, and ...www.w3.org/TR/1998/NOTE-HTMLThreading-0105 - 50k - [Cached](#) - [Similar pages](#)Parser Guide... The **parsable** objects, PICS Labels, machine-readable service descriptions, and, users, all ... are implemented in the following files: Labels are in CSLL.**html**; ...www.w3.org/PICS-client/User/Parsing.**html** - 8k - [Cached](#) - [Similar pages](#)[\[More results from www.w3.org \]](#)IPP Mail Archive: Re: IPP>MOD - HTML and IPP streams -Reply... Not a Print or a Cancel. The resulting **HTML** file will not be **parsable** by a program, but should be displayable by a browser and understandable to an end user. ...www.pwg.org/hypermall/ipp-pre980212/0860.**html** - 4k - [Cached](#) - [Similar pages](#)man page(1) manual page... D Data out. Generates a machine-**parsable** table of GeneMark output data that can be used as an input to further analytical tools. ...bioweb.pasteur.fr/docs/man/man/gm.1.**html** - 13k - [Cached](#) - [Similar pages](#)

Goooooooooooooogle ►

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)



html parsable daterange:2449718 **Search**

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google


[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 560 for "alert box" datrange:2449718-2451179. (0.56 seconds)

[winobj02.html](#)

... task. The alert method. alert (text). The alert displays Internet Explorers **alert box** with the text specified by text displayed in it. ...
[www.shorrock.u-net.com/axwinobj02.html](#) - 7k - [Cached](#) - [Similar pages](#)

Sponsored Links

Alert Box

Shop for Toys and Games!
 Compare & Buy from 1000's of Stores
[www.Shopping.com](#)

[See your message here...](#)

[Opening Windows and Frames](#)

... You can also use it for debugging purposes. First, the **alert box**. The **alert box** pops up in the middle of the screen with some text. It gives the user an OK box. ...
[www.acm.uiuc.edu/webmonkeys/javascript/opening_windows.html](#) - 6k
 - [Cached](#) - [Similar pages](#)

[The Web-Backlash of 1996 \(Jakob Nielsen's Alert Box April 1996\)](#)

The **Alert Box** for April 1996: The Web Backlash of 1996. By Jakob Nielsen, SunSoft Distinguished Engineer The permanent home of this ...
[www.sun.com/columns/alertbox/9604.html](#) - 7k - [Cached](#) - [Similar pages](#)

[The Alert Box June 1995](#)

The **Alert Box** for June 1995: History has a Lesson for HotJava. By Jakob Nielsen, SunSoft Distinguished Engineer Unfortunately, many ...
[www.sun.com/950523/columns/alertbox/](#) - 6k - [Cached](#) - [Similar pages](#)
 [[More results from www.sun.com](#)]

[javascript1r - Introduction](#)

... The following script would create a button on your page and arrange it so that a user's click would result in an **alert box** asking "Are you sure you want to ...
[userwww.sfsu.edu/~infoarts/technical/howto/wilson.javascript1r.html](#) - 16k - [Cached](#) - [Similar pages](#)

[Strings javascript ch 4](#)

... The second sample would generate an **alert box** for the user that printed "Are you sure?". `document.myform.inputfield1.value = "This should appear in the field".` ...
[userwww.sfsu.edu/~infoarts/technical/howto/wilson.javascript4r.html](#) - 35k - [Cached](#) - [Similar pages](#)
 [[More results from userwww.sfsu.edu](#)]

[\[PDF\] \[1998\] Moving Arden Syntax Outside of the \(Alert\) Box: A Paradigm ...](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)
 Moving Arden Syntax Outside of the **(Alert) Box**: A Paradigm for Supporting Multi-Step Clinical Protocols R. Matthew Sailors, ME, Richard L. Bradshaw, MS, Thomas ...
[www.amia.org/pubs/symposia/D005113.PDF](#) - [Similar pages](#)

[Menu Arguments](#)

... Alert Boxes. alert = a message to be displayed as the **alert box's** label. yescommand = a Prolog goal to be called if the user confirms. ...
[www.ltg.ed.ac.uk/Pleuk/current/node150.html](#) - 12k - [Cached](#) - [Similar pages](#)

[JavaScript Example 1](#)

... The **Alert box** is a variation of the confirm box. The **alert box** just displays the message. No option is given to cancel the request. **Alert Box**: ...
[www.coe.tamu.edu/~strader/JavaScript/Examples/example1.html](#) - 2k - [Cached](#) - [Similar pages](#)

[Educational JavaScripting : Getting Started with JavaScript](#)

... One way to provide information to the user is with the alert(text) function. This function brings up an **alert box** which displays the argument to the function. ...

www.math.grin.edu/~rebelsky/Tutorials/JavaScript/EdMedia97/basics.html - 7k -

[Cached](#) - [Similar pages](#)

Goooooooooooooogle ►

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)



"alert box" daterange:2449718-2451179 **Search**

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied?](#) [Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google